REMARKS

Please reconsider the present application in view of the above amendments and the following remarks. Applicant thanks the Examiner for the courtesies extended in the telephonic Examiner Interview of June 1, 2005 and for carefully considering this application.

Disposition of Claims

Claims 1, 3-6, 10, 12-15, and 17 are pending in the present application. By way of this reply, claims 1, 10, and 17 have been amended. Claims 1, 10, and 17 are independent. The remaining claims depend, directly or indirectly, from claims 1 and 10.

Claim Amendments

Independent claims 1 and 17 have been amended by way of this reply to indicate that a technique for optimizing predicated code during execution of the code comprises, in part, (i) testing a condition code associated with a conditional instruction, and (ii) writing Boolean data to a general register designated as a destination register based on the testing, the destination register corresponding to the conditional instruction and representing a predicate. Independent claim 10 has been amended to indicate that an apparatus for optimizing predicate code during execution of the code comprises, in part, (i) means for testing a condition code associated with a conditional instruction, and (ii) means for writing Boolean data to a general register designated as a destination register based on the testing, the destination register corresponding to the conditional instruction and representing a predicate. No new matter has been added by way of these amendments, as support for these amendments may be found, for example, in paragraphs [0052]-[0055] of the present application.

Rejection(s) under 35 U.S.C § 103

Claims 1, 3-4, 6, 10, 12-13, and 15

Claims 1, 3-4, 6, 10, 12-13, and 15 were rejected under 35 U.S.C. § 103(a) as being obvious over Intel® IA-64 "Architecture Software Developer's Manual (hereinafter "IA-64") in view of "A Framework for Balancing Control Flow and Predication" by Hwu *et al.* (hereinafter "Hwu"), further in view of U.S. Patent No. 6,637,026 issued to Chen (hereinafter "Chen"). Claims 1 and 10 have been amended in this reply to clarify the present invention recited. To the extent that this rejection may still apply to the amended claims, the rejection is respectfully traversed.

The present invention is directed to a method and apparatus for optimizing predicated code (*see* Specification, Abstract). As exemplarily discussed with reference to Figure 2, a processor in accordance with the present invention includes instructions to test a given conditional branch instruction to determine whether the branch will be taken. If the conditional branch is taken (true), then a general register specified as a destination register and corresponding to the conditional instruction is designated as 1 (true). In other words, the destination register of the conditional instruction is designated as 1 (true). Otherwise, a general register specified as a destination register and corresponding to the conditional instruction is designated as 0 (false). In other words, the destination register of the conditional instruction is designated as 0 (false) (*see* Specification, paragraph [0054]). By normalizing the data that was a condition code into general registers containing a true or false value, Boolean operations may be performed in combination with other predicates using normal logical operations. Thus, the need for special operations to deal with predicates is avoided (*see* Specification, paragraph [0055]).

Accordingly, amended claim 1 requires, in part, (i) testing a condition code associated with a conditional instruction, and (ii) writing Boolean data to a general register designated as a destination register based on the testing, the destination register corresponding to the conditional instruction and representing a predicate. Amended independent claim 10 requires that an apparatus for optimizing predicate code comprise, in part, (i) means for testing a condition code associated with a conditional instruction, and (ii) means for writing Boolean data to a general register designated as a destination register based on the testing, the destination register corresponding to the conditional instruction and representing a predicate. Amended independent claims 1 and 10 have been amended, as suggested by the Examiner in the telephonic Examiner interview of June 1, 2005, to indicate that optimizing predicated code occurs during execution of the predicated code.

IA-64, in contrast to the present invention, merely describes normal operations related to predicated execution ("predication"). For Example, in Section 8.5 of IA-64, predication is defined as the conditional execution of an instruction based on a qualifying predicate, where the qualifying predicate is a predicate register whose value determines whether the processor commits the results computed by an instruction (*see* IA-54, Section 8.5).

In section 10.3, IA-64 discusses problems related to control flow optimization. Specifically, IA-64 discusses problems related to several control flows converging at one point and for multiple control flows starting from one point. IA-64 notes that special instructions are provided to allow complex compound conditions to be reduced into a single condition with fewer computations than normally required. For example, IA-64 discusses parallel compare instructions to optimize instructions that have "and" and "or" operations. These instructions allow multiple "and/or" compare instructions to target the same instruction within a single instruction group, thus allowing a compound conditional to be resolved in a single cycle (*see* IA-

64, Section 10.3, *e.g.*, paragraph beginning with "To reduce the cost of compound conditionals...").

Specifically, IA-64 states that for the usage model to work properly, during any given execution of the code, all instructions that target a given predicate register must either: a) write the same value (0 or 1), or b) not write to the target register at all. IA-64 is clear in implying that values are written to a predicate register when testing a conditional code. Further, IA-64 implies that the target registers are separate registers written to and initialized prior to execution of an instruction (*e.g.*, compare, "cmp") (*see* IA-64, Section 10.3, *e.g.*, section beginning with "For this usage model to work properly..."). *IA-64 is completely silent with respect to writing Boolean data to a general register designated as a destination register based on testing a condition code associated with a conditional instruction, where the destination register corresponds to the conditional instruction and represents a predicate.* Accordingly, IA-64 fails at least to disclose the above limitations of the claimed invention.

Hwu, like IA-64 discussed above, fails to disclose all the limitations of the claimed invention or supply that which IA-64 lacks. Hwu is directed to addressing what instructions in a program should be if-converted, as well as when if-conversion should be applied (*see* Hwu, abstract). Hwu discusses issues faced in forming hyperblocks, which are structures that facilitate optimization and scheduling for predicated architectures. Specifically, a hyperblock is a set of predicated basic blocks in which control may enter only from the top, but may exit from one or more locations (*see* Hwu, Section 2.1, first paragraph). In order to create more efficient predicated code, Hwu discusses the adjustment of hyperblocks during scheduling, which is known as partial reverse if-conversion (*see* Hwu, Section 3, fifth paragraph). *However, Hwu is completely silent with respect to writing Boolean data to a general register designated as a destination register based on testing a condition code associated with a conditional instruction,*

where the destination register corresponds to the conditional instruction and represents a predicate. Accordingly, like IA-64 discussed above, Hwu fails at least to disclose those limitations of the claimed invention not disclosed in IA-64.

Chen, like IA-64 and Hwu discussed above, fails to disclose all the limitations of the claimed invention or supply that which IA-64 and Hwu lack. Chen is directed to a technique for inserting an alert instruction to alert a global register allocator to map particular virtual predicates into the same physical registers (*see* Chen, abstract). Chen eliminates predicate generating instructions by finding path-equivalent blocks of instructions in the future flow of the program and merging the instructions (*see* Chen, col. 2, lines 35-39). *However, Chen is completely silent with respect to writing Boolean data to a general register designated as a destination register based on testing a condition code associated with a conditional instruction, where the destination register corresponds to the conditional instruction and represents a predicate.* Accordingly, Chen fails at least to disclose those limitations of the claimed invention not disclosed in IA-64 and Hwu.

In view of the above, IA-64, Hwu, and Chen, whether considered separately or in any combination, fail to show or suggest the present invention as recited in amended independent claims 1 and 10. Thus, amended independent claims 1 and 10 are patentable over IA-64, Hwu, and Chen. Dependent claims are allowable for at least the same reasons. Accordingly, withdrawal of this rejection is respectfully requested.

Claims 5, 14, and 17

Claims 5, 14, and 17 were rejected under 35 U.S.C. § 103(a) as being obvious over IA-64 in view of Hwu and Chen, and further in view of U.S. Patent No. 5,999,738 issued to Schlansker *et al.* (hereinafter "Schlansker"). Claims 1, 10, and 17 have been amended in this

reply to clarify the present invention recited. To the extent that this rejection may still apply to the amended claims, the rejection is respectfully traversed.

As discussed above, amended claims 1 and 17 require, in part, (i) testing a condition code associated with a conditional instruction, and (ii) writing Boolean data to a general register designated as a destination register based on the testing, the destination register corresponding to the conditional instruction and representing a predicate. Amended independent claim 10 requires that an apparatus for optimizing predicate code comprise, in part, (i) means for testing a condition code associated with a conditional instruction, and (ii) means for writing Boolean data to a general register designated as a destination register based on the testing, the destination register corresponding to the conditional instruction and representing a predicate. Amended independent claims 1, 10, and 17 have been amended, as suggested by the Examiner in the telephonic Examiner interview of June 1, 2005, to indicate that optimizing predicated code occurs during execution of the predicated code.

As discussed above, IA-64, Hwu, and Chen, whether considered separately or in any combination, fail to show or suggest the above limitations of amended independent claims 1, 10, and 17. Schlansker fails to provide that which IA-64, Hwu, and Chen lack. Schlansker is directed to a technique for flexibly scheduling a code sequence by generating a set of instructions for determining a fully-resolved predicate for each of a set of non-speculative instruction in the code sequence. This allows a non-speculative instruction to be executed before any other non-speculative instruction (*see* Schlansker, abstract). *However, Schlansker is completely silent with respect to writing Boolean data to a general register designated as a destination register based on testing a condition code associated with a conditional instruction, where the destination register corresponds to the conditional instruction and represents a*

predicate. Accordingly, Schlansker fails at least to disclose those limitations of the claimed invention not disclosed in IA-64, Hwu, and Chen.

Further, Applicant notes that the various combinations of one or more of four references have been used in rejecting the claims of the present application. The purported reconstruction of the claimed invention by reliance on this number of references ranging from, for example, a microprocessor software developer's manual (IA-64) to a technique for flexible scheduling of a code sequence (U.S. Patent No. 5,999,738) is not appropriate. It is abundantly clear that the Examiner, using the present application as a guide, has selected isolated features of the various relied-upon references to arrive at the limitations of the claimed invention. Use of the present application as a "road map" for selecting and combining prior art disclosures is wholly improper. *See Interconnect Planning Corp. v. Feil*, 774 F.2d 1132 (Fed. Cir. 1985) (stating that "[t]he invention must be viewed not with the blueprint drawn by the inventor, but in the state of the art that existed at the time"); *In re Fritch*, 972 F.2d 1260 (Fed. Cir. 1992) (stating that "it is impermissible to use the claimed invention as an instruction manual or 'template' to piece together the teachings of the prior art so that the claimed invention is rendered obvious .... This court has previously stated that 'one cannot use hindsight reconstruction to pick and choose among isolated disclosures in the prior art to deprecate the claimed invention.'"); *In re Wesslau*, 353 F.2d 238 (C.C.P.A. 1965) (stating that "it is impermissible within the framework of section 103 to pick and choose from any one reference only so much of it as will support a given position, to the exclusion of other parts necessary to the full appreciation of what such reference fairly suggests to one of ordinary skill in the art").

Further, Applicant notes that there is no motivation to combine the cited references. The Examiner cannot combine prior art references to render a claimed invention obvious by merely showing that all the limitations of the claimed invention can be found in the prior art

references. There must be a suggestion or motivation to combine the references *within the prior art references themselves*. In other words, regardless of whether prior art references can be combined, there must an indication within the prior art references expressing desirability to combine the references. *In re Mills*, 916 F.2d 680 (Fed. Cir. 1990) (emphasis added). Further, the present application *cannot be used a guide* in reconstructing elements of prior art references to render the claimed invention obvious. *In re Vaeck*, 947 F.2d 488 (Fed. Cir. 1991) (emphasis added).

One skilled in the art would not be motivated by IA-64, Hwu, or Chen, which are completely silent with respect to eliminating a guarding predicate of an unsafe instruction by compensation, to incorporate the teachings of Schlansker without the present application as a guide. The Examiner assumes that analysis of dependency to avert from using inefficient instructions implies using compensation code to stand for predicate code. However, IA-64, Hwu, and Chen provide no motivation to use compensation to eliminate a guarding predicate of an unsafe instruction. Thus, one skilled in the art would not be motivated by IA-64, Hwu, or Chen to incorporate the teaching of Schlansker, which is directed to a technique for flexible scheduling of a code sequence. Further, Schlansker is silent with respect to (i) testing a condition code associated with a conditional instruction, and (ii) writing Boolean data to a general register designated as a destination register based on the testing, the destination register corresponding to the conditional instruction and representing a predicate. Thus, there is no motivation to combine the cited references.

In view of the above, IA-64, Hwu, Chen, and Schlansker (i) are not properly combinable, and (ii) whether considered separately or in any combination, fail to show or suggest the present invention as recited in amended independent claims 1, 10, and 17. Thus, amended independent claims 1, 10, and 17 are patentable over IA-64, Hwu, Chen, and

Schlansker. Dependent claims are allowable for at least the same reasons. Accordingly, withdrawal of this rejection is respectfully requested.

Conclusion

Applicant believes this reply is fully responsive to all outstanding issues and places the present application in condition for allowance. If this belief is incorrect, or other issues arise, the Examiner is encouraged to contact the undersigned or his associates at the telephone number listed below. Please apply any charges not covered, or any credits, to Deposit Account 50-0591 (Reference Number 03226/037001; P5009).

Dated: July 18, 2005

Respectfully submitted,

By #45,079
Wasif H. Qureshi THOMAS SCHERER
Registration No.: 51,048
OSHA • LIANG LLP
1221 McKinney St., Suite 2800
Houston, Texas 77010
(713) 228-8600
(713) 228-8778 (Fax)
Attorney for Applicant

103602_1